XAL Report

Thomas Pelaia II

EPICS Collaboration Meeting 2004

May 5, 2004

XAL Collaboration

- Chris Allen (LANL)
- Chungming P. Chu (ORNL)
- John Galambos (ORNL) - **Group Leader**
- Wolf-Dieter Klotz (ESRF)
- Craig McChesney (LANL)
- Thomas Pelaia (ORNL)
- Andrei Shishlo (ORNL)
- Cosylab group

Acknowledgments

- Jeff Patton (ORNL)
- Nikolay Malitsky (BNL)
- Peregrine McGehee (LANL)
- Nick Pattengale (formerly at LANL)
- Dan Ottavio
- EPICS Community

Background

- XAL is a collection of tools written in pure Java and designed to support application development for accelerator physics
  - interface for Java Channel Access
  - accelerator modeling
  - generic tools
- The application framework allows for rapid development of applications with a common look and feel

Sampling of XAL tools

- **XAL includes many tools to assist in development. Here are just a few:**
  - **application framework**
  - **messaging framework**
  - **services framework**
  - **high performance formula interpreter**
  - **plotting framework**
  - **logbook access**
  - **database access**
  - **optimization framework**

Application Framework Status

- Application framework adopted for all new applications
- Rapid development of quality applications
- Applications inherit many common user interface behaviors
- About two dozen applications have been written
- Physicists are now writing applications
- Proven flexibility to support a wide variety of applications

Application Framework Improvements

- Support for multi-document applications (i.e. different kinds of documents)
- Support for frames in HTML help
- Several documents can be opened upon launch
- Every application advertises itself on the local network
- More support for customizing documents
- Improved user feedback (e.g. warnings)
- Many more new features

Existing Application Categories

- Orbit management
- Running scans
- Machine state recovery
- Logging
- Modeling
- Diagnostics
- Utilities

XAL Service Framework

- XAL Service Framework introduced
- Allows services to be dynamically offered
- Supports open standards
- Wraps protocols to support future changes
- Easy to use
- Good performance
- Two core technologies:
  - Service advertisement and discovery
  - Handling requests

Service discovery

- **Uses Apple Rendezvous zero configuration**
  - **open source and C code is available for several platforms**
  - **XAL uses the Java version (JRendezvous)**
  - **Reliable and simple to use**
  - **Allows services to be broadcast and discovered**
  - **Works across many protocols (TCP, Firewire, ...)**
  - **Does not require a central server**
  - **Widely adopted by industry (Apple, HP, Xerox, Canon, Sybase, TiVo, ...) in a wide variety of high performance applications (grid clustering, RAID, printers, ...)**

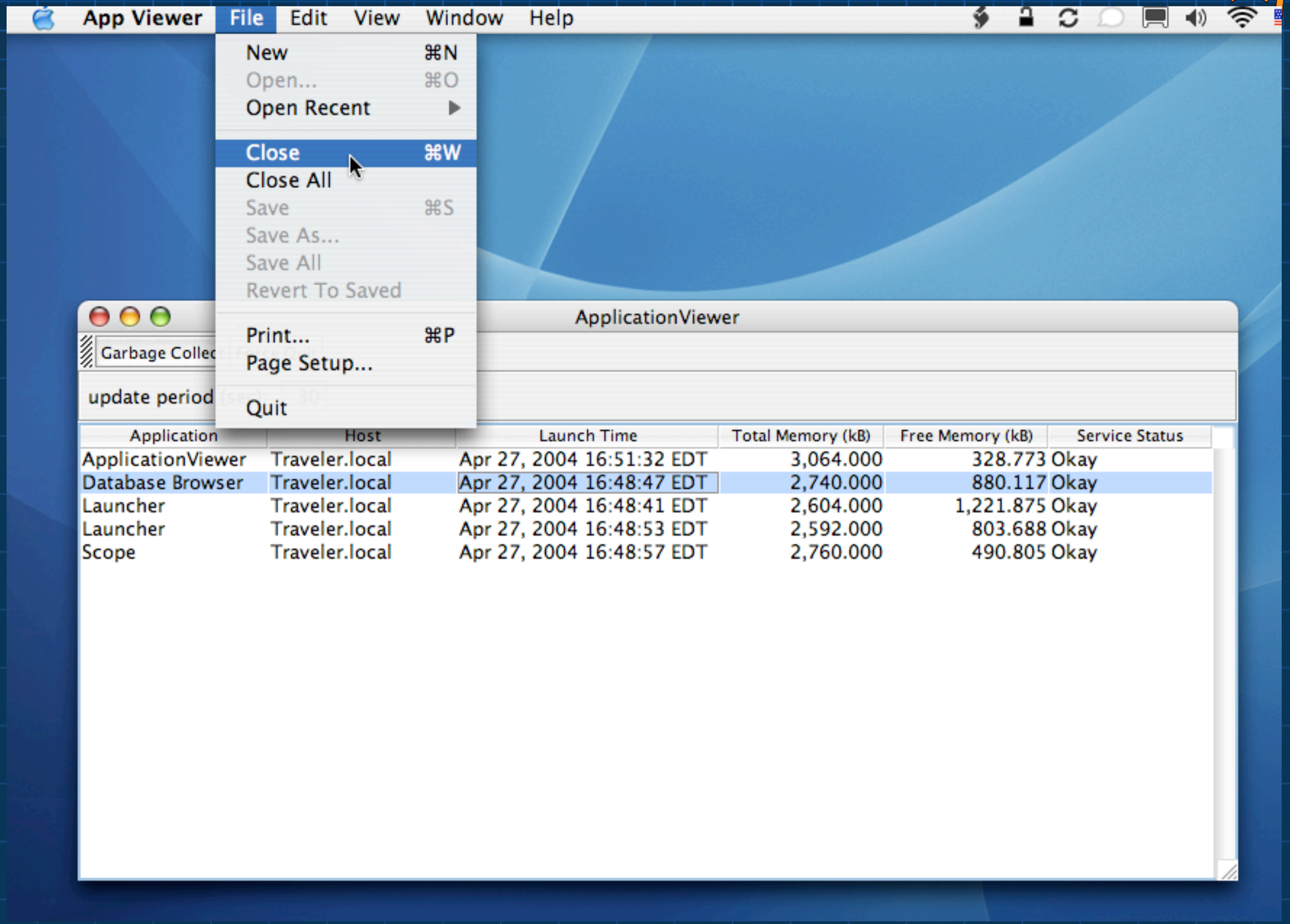
Service Request Handling

- **Uses XML-RPC for service request handling**
  - **Cross platform and open source**
  - **XAL uses the Java implementation**
  - **Simple to use**
  - **Fast**

Applications provide services

- Every XAL application automatically provides a service
- Applications advertise their existence and provide basic actions
  - Provide start time
  - Provide host
  - Provide memory use
  - Allow garbage collection
  - Allow termination

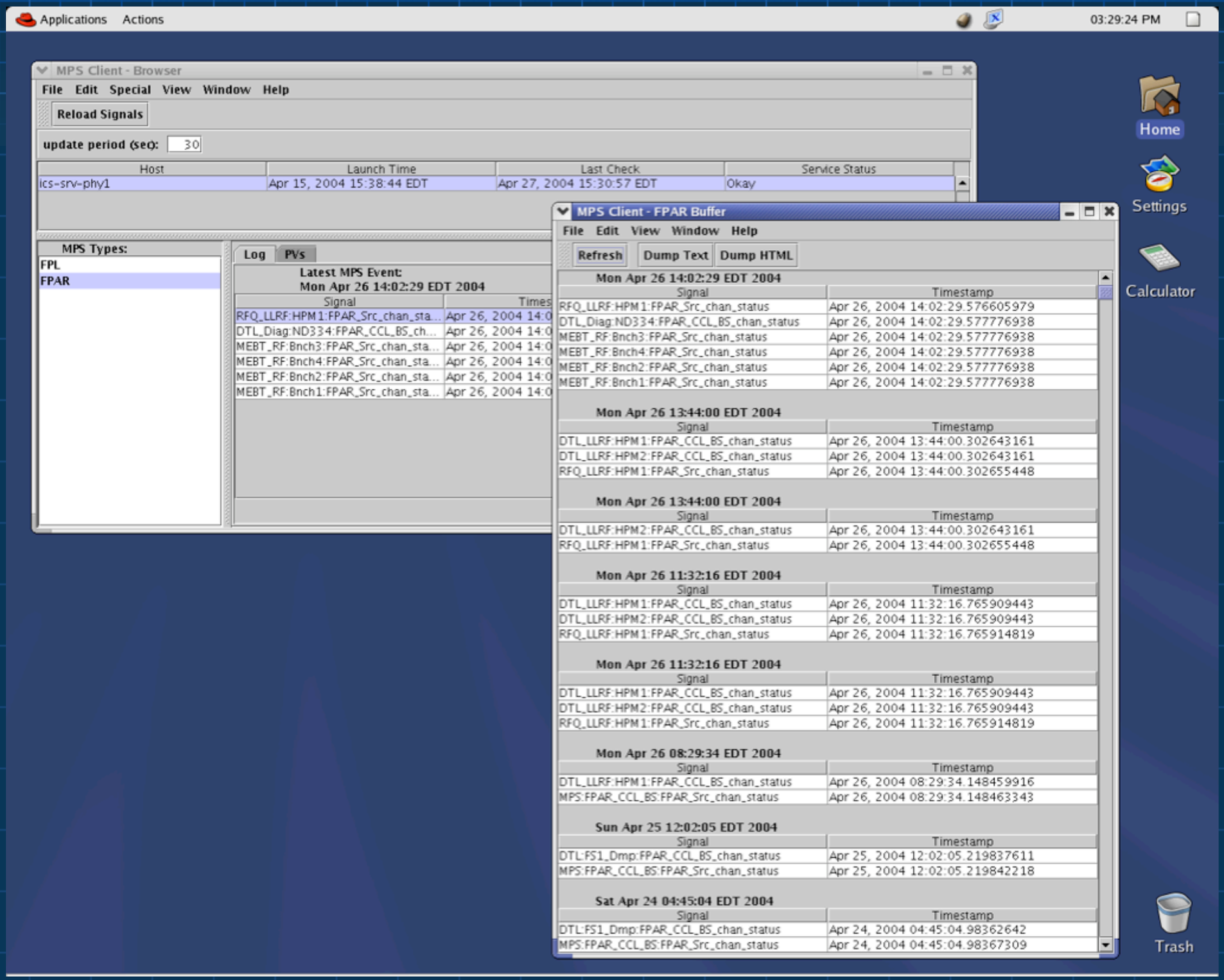
Application Viewer



Pure Services

- **Headless Java processes that run as true Unix services**
- **Presently have two pure services**
  - **PV Logger**
    - periodically log a set of PV states to the database
  - **MPS Service**
    - Monitor MPS faults

MPS Client


Applications Actions
03:29:24 PM
MPS Client - Browser
File Edit Special View Window Help
Reload Signals
update period (sec): 30
Host | Launch Time | Last Check | Service Status
ics-srv-phy1 | Apr 15, 2004 15:38:44 EDT | Apr 27, 2004 15:30:57 EDT | Okay
MPS Types:
FPL
FPAR
Log PVs
Latest MPS Event:
Mon Apr 26 14:02:29 EDT 2004
Signal | Times
RFQ_LLRF:HPM1:FPAR_Src_chan_sta... | Apr 26, 2004 14:0
DTL_Diag:ND334:FPAR_CCL_BS_ch... | Apr 26, 2004 14:0
MEBT_RF:Bnch3:FPAR_Src_chan_sta... | Apr 26, 2004 14:0
MEBT_RF:Bnch4:FPAR_Src_chan_sta... | Apr 26, 2004 14:0
MEBT_RF:Bnch2:FPAR_Src_chan_sta... | Apr 26, 2004 14:0
MEBT_RF:Bnch1:FPAR_Src_chan_sta... | Apr 26, 2004 14:0
MPS Client - FPAR Buffer
File Edit View Window Help
Refresh Dump Text Dump HTML
Mon Apr 26 14:02:29 EDT 2004
Signal | Timestamp
RFQ_LLRF:HPM1:FPAR_Src_chan_status | Apr 26, 2004 14:02:29.576605979
DTL_Diag:ND334:FPAR_CCL_BS_chan_status | Apr 26, 2004 14:02:29.577776938
MEBT_RF:Bnch3:FPAR_Src_chan_status | Apr 26, 2004 14:02:29.577776938
MEBT_RF:Bnch4:FPAR_Src_chan_status | Apr 26, 2004 14:02:29.577776938
MEBT_RF:Bnch2:FPAR_Src_chan_status | Apr 26, 2004 14:02:29.577776938
MEBT_RF:Bnch1:FPAR_Src_chan_status | Apr 26, 2004 14:02:29.577776938
Mon Apr 26 13:44:00 EDT 2004
Signal | Timestamp
DTL_LLRF:HPM1:FPAR_CCL_BS_chan_status | Apr 26, 2004 13:44:00.302643161
DTL_LLRF:HPM2:FPAR_CCL_BS_chan_status | Apr 26, 2004 13:44:00.302643161
RFQ_LLRF:HPM1:FPAR_Src_chan_status | Apr 26, 2004 13:44:00.302655448
Mon Apr 26 13:44:00 EDT 2004
Signal | Timestamp
DTL_LLRF:HPM2:FPAR_CCL_BS_chan_status | Apr 26, 2004 13:44:00.302643161
RFQ_LLRF:HPM1:FPAR_Src_chan_status | Apr 26, 2004 13:44:00.302655448
Mon Apr 26 11:32:16 EDT 2004
Signal | Timestamp
DTL_LLRF:HPM1:FPAR_CCL_BS_chan_status | Apr 26, 2004 11:32:16.765909443
DTL_LLRF:HPM2:FPAR_CCL_BS_chan_status | Apr 26, 2004 11:32:16.765909443
RFQ_LLRF:HPM1:FPAR_Src_chan_status | Apr 26, 2004 11:32:16.765914819
Mon Apr 26 11:32:16 EDT 2004
Signal | Timestamp
DTL_LLRF:HPM1:FPAR_CCL_BS_chan_status | Apr 26, 2004 11:32:16.765909443
DTL_LLRF:HPM2:FPAR_CCL_BS_chan_status | Apr 26, 2004 11:32:16.765909443
RFQ_LLRF:HPM1:FPAR_Src_chan_status | Apr 26, 2004 11:32:16.765914819
Mon Apr 26 08:29:34 EDT 2004
Signal | Timestamp
DTL_LLRF:HPM1:FPAR_CCL_BS_chan_status | Apr 26, 2004 08:29:34.148459916
MPS:FPAR_CCL_BS:FPAR_Src_chan_status | Apr 26, 2004 08:29:34.148463343
Sun Apr 25 12:02:05 EDT 2004
Signal | Timestamp
DTL:FS1_Dmp:FPAR_CCL_BS_chan_status | Apr 25, 2004 12:02:05.219837611
MPS:FPAR_CCL_BS:FPAR_Src_chan_status | Apr 25, 2004 12:02:05.219842218
Sat Apr 24 04:45:04 EDT 2004
Signal | Timestamp
DTL:FS1_Dmp:FPAR_CCL_BS_chan_status | Apr 24, 2004 04:45:04.98362642
MPS:FPAR_CCL_BS:FPAR_Src_chan_status | Apr 24, 2004 04:45:04.98367309
Home
Settings
Calculator
Trash

Define a Service Interface

```java
/*
 * Greeting.java
 *
 * Created on Wed Aug 27 10:48:25 EDT 2003
 *
 * Copyright (c) 2003 Spallation Neutron Source
 * Oak Ridge National Laboratory
 * Oak Ridge, TN 37830
 */

package gov.sns.tools.services.samples;


/**
 * Greeting is a sample service demonstrating a simple interface that can be served using the ServiceDirectory.
 *
 * @author  tap
 */
public interface Greeting {
    /**
     * Generate a greeting incorporating the receiver.
     * @param receiver The receiver to address in the greeting.
     * @return A greeting.
     */
    public String sayHelloTo(String receiver);


    /**
     * Add two numbers and return the result.
     * @param x An addend
     * @param y An augend
     * @return The sum of x and y.
     */
    public double add(double x, double y);
}
```

Register a Service Provider

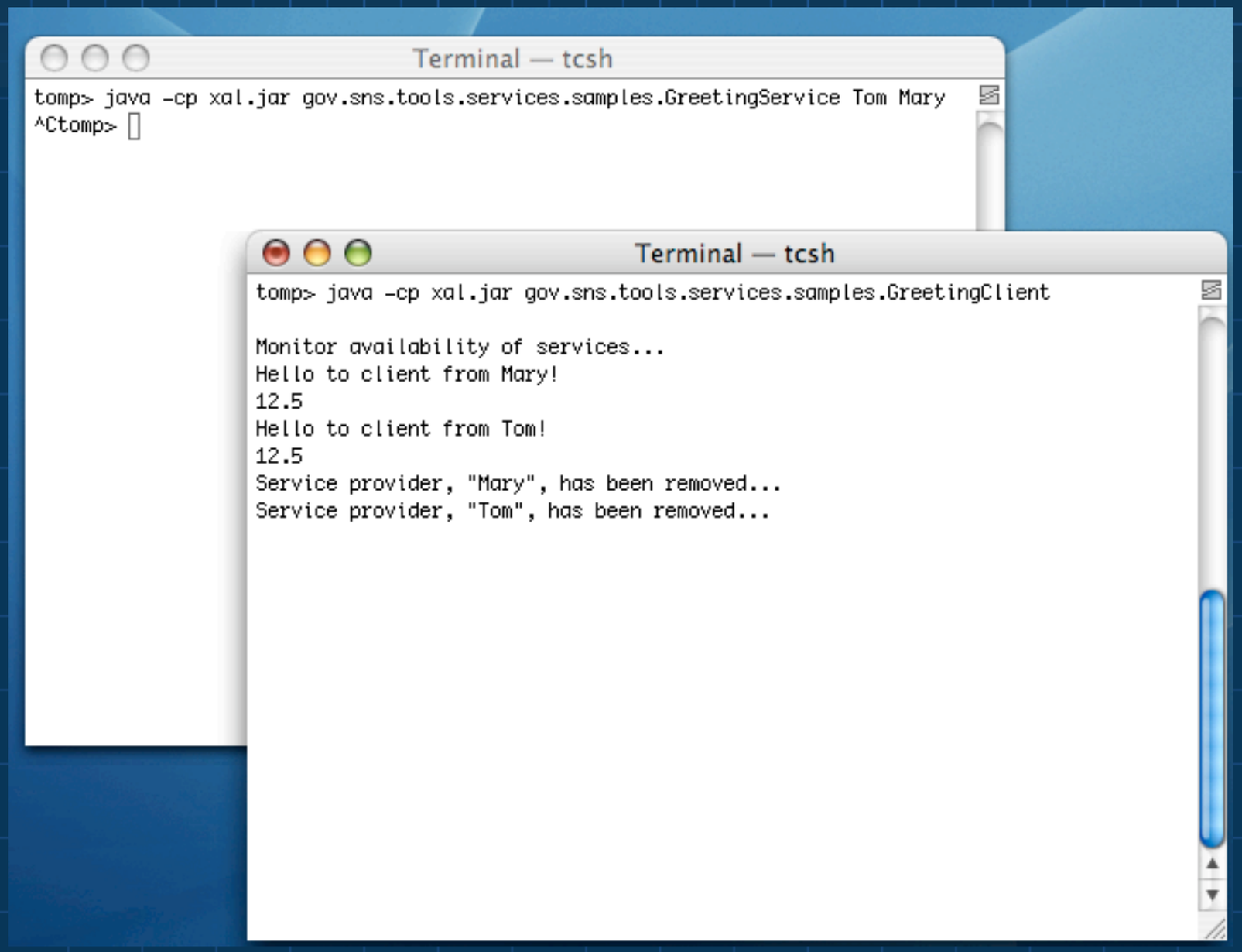
```java
/*
 * GreetingService.java
 *
 * Created on Wed Aug 27 11:38:29 EDT 2003
 *
 * Copyright (c) 2003 Spallation Neutron Source
 * Oak Ridge National Laboratory
 * Oak Ridge, TN 37830
 */

package gov.sns.tools.services.samples;

import gov.sns.tools.services.*;


/**
 * GreetingService is a sample service provider implementing the Greeting service.
 *
 * @author  tap
 */
public class GreetingService implements Greeting {
    final protected String _identity;


    /**
     * GreetingService constructor registers the greeting service.
     * @param identity The identity of the this greeting service provider.
     */
    public GreetingService(String identity) {
        _identity = identity;
        ServiceDirectory.defaultDirectory().registerService(Greeting.class,
               _identity, this);
    }
```

```java
    /**
     * Generate a greeting incorporating the receiver.
     * @param receiver The receiver to address in the greeting.
     * @return A greeting.
     */
    public String sayHelloTo(String receiver) {
        return "Hello to " + receiver + " from " + _identity + "!";
    }


    /**
     * Add two numbers and return the result.
     * @param x An addend
     * @param y An augend
     * @return The sum of x and y.
     */
    public double add(double x, double y) {
        return x + y;
    }


    /**
     * Launch the Greeting service.
     * @param args The identities of each service to provide.
     */
    static public void main(String[] args) {
        for ( int index = 0 ; index < args.length ; index++ ) {
            new GreetingService(args[index]);
        }
    }
}
```

Create a Client

```java
/*
 * GreetingClient.java
 *
 * Created on Wed Aug 27 10:51:37 EDT 2003
 *
 * Copyright (c) 2003 Spallation Neutron Source
 * Oak Ridge National Laboratory
 * Oak Ridge, TN 37830
 */

package gov.sns.tools.services.samples;

import gov.sns.tools.services.*;


/**
 * GreetingClient is a sample client that demonstrates how to use the ServiceDirectory
 * to lookup a service and send messages to it.
 *
 * @author  tap
 */
public class GreetingClient {
    /**
     * Launch a Greeting client to find listen for greeting services and make requests on them.
     * @param args The arguments to the client (currently ignored)
     */
    static public void main(String[] args) {
        // Listen for services being added and removed.
        System.out.println("\nMonitor availability of services...");
        ServiceDirectory.defaultDirectory().addServiceListener(Greeting.class, new ServiceListener() {
            public void serviceAdded(ServiceDirectory directory, ServiceRef serviceRef) {
                Greeting proxy = (Greeting)directory.getProxy(Greeting.class, serviceRef);
                System.out.println( proxy.sayHelloTo("client") );
                System.out.println( proxy.add(5.1, 7.4) );
            }

            public void serviceRemoved(ServiceDirectory directory, String type, String name) {
                System.out.println("Service provider, \"" + name + "\", has been removed...");
            }
        });
    }
}
```

Running the Sample Service



Future Directions

- Build many more applications
- Build more services
- Add more tools
- Continue to enhance existing applications
- Continue to add support for the Ring
- Migrate to JCA 2.1